

FIFA-2022

FIFA World Cup Qatar 2022 Event Query Application



FIFA WORLD CUP
Qatar2022

License MPL Python 3.9.13 PyQt5 5.15.4

Content

[Introduction](#)

[Structure](#)

[Installation](#)

[Basic Configuration](#)

[Document Conversion Method](#)

[Webcrawler Configuration](#)

[Project packaged as an application](#)

[Call for Contributions](#)

[Document](#)

Introduction

In this project, I created a graphical interface query system for the World Cup tournament. I used **PyQt5** library module for graphical interface design, **socket** library for network connection and data transfer, **sqlite3** library for data storage, **re** library for input format correctness, **os** library for file path reading and **sys** library for process exit. I also use basic common programming methods such as **multi-threading**, **object-oriented programming**, **string manipulation**, and **custom functions**.

I also designed a web crawler file, using the **webdriver** module in the **selenium** library, while using the **pyautogui** library that can control the cursor movement, with the **re** library to get the content of the web page and the time library to simulate the response time, these do assist in crawling.

Since the web crawler requires third-party libraries and corresponding drivers, the environment is more complicated to configure, so I separate the web crawler files from the server-side files, and the crawled data is manually added into the database by me.

Structure

```
└─FIFA-2022 :: Root Directory
  | client.py :: Client Code
  | README.md :: Project Description
  | server.py :: Server Code
  |
  └─Document :: Project Documentation
    | Introduction.pdf :: Basic Introduction
    | Report.pdf :: Design Report
    |
    └─Element :: Image Elements
      | 2022_FIFA_world_Cup.svg
      |
      └─Tournament :: Race Data Acquisition
        | web_crawler.py :: Web Crawler
        |
        └─UI :: User Interface Design
          administrator.py :: Administrator Interface
          Image.py :: Machine code data for the background image of each
sector
          main.py :: Program Initial Screen
          register.py :: User Registration Interface
          user.py :: Race Tracking Interface
```

Installation

Basic Configuration

To install PyQt5, you can use the follow command in terminal:

```
pip install PyQt5
```

If you want to design UI more convenient, you can also install PyQt5-tools:

```
pip install PyQt5-tools
```

If you have a higher version of python, you may not be able to use the tools.

Document Conversion Method

Then you need to configure the following three tools: **Qt Designer**、**PyRec**、**PyUIC**. If you do not like to do this, just check them installation path and use them when you need.

If you want to convert **.ui** file to **.py** file, you can enter the current directory of the ui file then input the following in the **PowerShell** command line:

```
pyuic5 $FileName$ -o $FileNameWithoutExtension$.py
```

For example:

```
pyuic5 interface.ui -o interface.py
```

If you want to convert *.qrc* file to *.py* file, you can enter the current directory of the qrc file then input the following in the *PowerShell* command line:

```
pyrcc5 $FileName$ -o $FileNameWithoutExtension$.py
```

For example:

```
pyrcc5 Source.qrc -o Image.py
```

Webcrawler Configuration

In order to run the crawler file, you need to additionally install the following.

```
pip install pyautogui  
pip install selenium
```

Then you also need to download a browser driver file. For example, the Edge download path is [Microsoft Edge WebDriver - Microsoft Edge Developer](#).

Download the version you choose, unpack it and put the files in the *Scripts* folder of the Python installation directory.

Project packaged as an application

To package the entire project file as an application *.exe* file, we can use *PyInstaller*, a third-party library.

```
pip install pyinstaller
```

Client-side and server-side packaging can be done by executing the following code using the console in the project's directory.

```
pyinstaller -D -w client.py
```

```
pyinstaller -D server.py
```

Call for Contributions

The project welcomes your expertise and enthusiasm!

Writing code isn't the only way to contribute to NumPy. You can also:

- review pull requests

- help us stay on top of new and old issues
- develop tutorials, presentations, and other educational materials
- help with outreach and onboard new contributors
- write grant proposals and help with other fundraising efforts

Our preferred channels of communication are all public, but if you'd like to speak to me in private first, contact at xiashj21@lzu.edu.cn or on Slack (write xiashj21@lzu.edu.cn for an invitation).

If you are new to contributing to open source, [this guide](#) helps explain why, what, and how to successfully get involved.

Document

- [Introduction](#)
- [Design Report](#)